

A Method and a Bridge for Coupling a Server and a Client of Different Object Types

The invention relates to a bridge for coupling a client to a server computer and to a corresponding method, computer system and a computer program.

From Statutory Invention Registration U.S. H1,895 an application provider and method for communication is provided for communication of an application provider between a first sub-system and a second sub-system of a common node.

For example, the application provider may be a mobile application provider, the first and second sub-systems may be a home location register and a visitor location register, and the common node may be a wireless telecommunication system. The system is implemented by means of an object request broker (ORB) such as the common request broker architecture (CORBA) defined by the object management group (OMG). A resource manager application serves as a "bridge" or "conduit" between a call processor and a resource assembly.

From U.S. 6,094,688 a system is known for modular application collaboration including filtering at the source and proxy execution of compensating transactions to conserve server resources. This system provides interoperability between applications including a plurality of connectors for communicating with a like plurality of applications and an interchange server.

The interchange server includes an application collaboration module and service module. The services module transfers messages between connectors and the application collaboration module. The application collaboration defines the interoperability between two or more applications. The interchange server service module includes a transaction service and an error service. Transactions are executed in the application collaboration module.

From U.S. 6,085,220 an enterprise interaction hub for managing an enterprise web system is known. The enterprise interaction hub includes a number of layers that interact to manage an enterprise web system.

An interaction layer receives requests to the enterprise web system and returns responsive web pages. A presentation layer is coupled to the interaction layer and generates the responsive web pages. A business layer is coupled to the presentation layer and provides business logic for use by the presentation layer in generating the responsive web pages.

An integration layer is coupled to the business layer and interfaces with existing legacy data to provide the legacy data to the business layer. A trend collection layer monitors and accumulates historical information from the interaction layer, the presentation layer, the business layer and the integration layer. The trend collection layer also stores the historical information in a trend database.

A profile database, accessible by the presentation layer and the business layer, stores profile data, including data mined from the trend database, that characterizes individual user access to the enterprise web system. The profile data is used by the

presentation layer and the business layer to provide customized dynamic content in the generated web pages.

From U.S. 6,085,030 a network component server is known. The component server architecture enables consumer nodes of a computer network to interact with heterogeneous software components and services distributed throughout the network, as well as network devices and data.

Distributed interactions between a consumer and a heterogeneous software is achieved, in part, by registering and locating the component and services. An object neutral global component registry with access controls of the architecture interoperates with the component management service to transparently ensure proper administration, of indication and run-time binding access to components offered in response to requests from applications executing on the consumer nodes.

The architecture is implemented on a component server node of the network that is configured to communicate with the consumer, i.e., client, nodes in client-server computing arrangements. That is, the component registry of the component server node responds to a consumer application request by locating a heterogeneous component for the consumer. The registry offers this component to the consumer by providing an appropriate interface between the object module of the software component. This registry is organized as a plurality of cooperating storage entities including a description repository, an offer repository, an interface adapter repository and an object factory repository.

A component server architecture is provided to interact with heterogeneous software components and services distributed

throughout the network, as well as with network devices and data. Distributed interaction between a consumer and heterogeneous software is achieved, in part, by registering and locating the components and services.

An object-neutral global component registry with access controls of the architecture interoperates with a component management service (CMS) to transparently ensure proper administration, authentication and run-time binding access to components offered in response to requests from applications executing on the consumer nodes. The architecture is implemented on a component server node of the network that is configured to communicate with the consumer, i.e., client, nodes in client-server computing arrangements. That is, the component registry of the component server node responds to a consumer application request by locating a heterogeneous component for the consumer. The registry offers this component to the consumer by providing an appropriate interface between the object module of the consumer and the object module of the software component.

This registry is preferably organized as a plurality of cooperating storage entities including a description repository, an offer repository, an interface adapter repository and an object factory repository.

From U.S. 5,913,061 a modular application collaborator is known for providing inter-operability between applications including a plurality of connectors for communicating with a like plurality of applications and an interchange server. The interchange server includes an application collaboration module and a service module.

The service module transfers messages between connectors and the application collaboration module. The application collaboration module defines the inter-operability between two or more applications and includes a trigger and a transaction responsive to the trigger. The trigger is activated upon receipt of data from one or more connectors resulting in the transaction delivering data to one or more connectors for transfer to an associated application.

From U.S. 6,115,646 a dynamic and generic process automation system is known. This includes a generic object-oriented process automation engine that provides workflow management services in a heterogeneous distributed computing environment. The system consists of three major parts: a build time part used to capture and store process definitions, and to request the enactment of a process; a run time part used to schedule, execute , and monitor the requested process; a CORBA to plug-in-software applications needed to execute processes and to allow interactions among the system components. This system is event-driven and provides scheduling and resource allocations schemes.

The system can incorporate a trader service which provides a "yellow pages" for objects; it allows objects to publicize their services and allow clients to find them based on upon which services the client needs. For example, a resource allocator can use the service to find a resource that satisfies a set of agents constrains.

The above referenced prior art systems and methods mostly rely on object oriented software technologies, such as CORBA. CORBA is a vendor-independent architecture and infrastructure that computer applications use to work together over networks. In

CORBA, every object instance has its own unique object reference, an identifying electronic token.

Clients use the object references to direct their invocations, identifying to the object request broker the exact instance they want to invoke. The client acts as if its invoking an operation on the object instance, but it is actually invoking on the IDL stub which acts as a proxy. Passing through the stub on the client side, the invocation continues through the object request broker, and the skeleton on the implementation side, to get to the object where it is executed on the server.

The object request broker provides a mechanism for transparently communicating client requests to target object implementations. The object request broker simplifies distributed programming by decoupling the client from the details of the method invocations. This makes client requests appear to be local procedure calls. When a client invokes an operation, the object request broker is responsible for finding the object implementation, transparently activating it if necessary, delivering the request to the object and returning any response to the caller.

The IDL stubs and skeletons serve as the "glue" between the client and the server applications, respectively, and the object request broker. The transformation between IDL definitions and the target programming language is automated by a IDL compiler. The use of a compiler reduces the potential for inconsistencies between client stubs and server skeletons and increases opportunities for automated compiler optimizations.

Another important object oriented technology is Enterprise Java Beans. Enterprise Java Beans provides a framework for components

that can be "plugged in" to a server, thereby extending that service functionality. Client programs execute methods on remote Enterprise Java Beans by way of an Enterprise Java Beans object. The Enterprise Java Beans object implements the "remote interface" of the EJB component on the server.

A common disadvantage of the CORBA and Enterprise Java Beans object technologies is that corresponding applications being available in an company's intranet are not accessible through a firewall from the internet.

For this and other reasons companies recently have started to use the Simple Object Access Protocol (SOAP) as the object oriented technology of choice as (SOAP) is firewall compatible. SOAP is a protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of applications - defined data types, and a convention for representing remote procedure calls and responses. SOAP can be used in combination within an HTTP extension framework.

Fig. 1 shows an example of a prior art network. The network includes internet 1 and intranet 2 which are separated by firewall 3. A SOAP client 4 is coupled to SOAP server 5 via internet 1.

Within the companies' intranet 2 a CORBA client 6 and an EJB client 7 are coupled to a CORBA server 8 and an EJB server 9, respectively.

Because of the tight coupling between the CORBA and EJB clients 6, 7 and their respective servers 8, 9 the applications of servers 8 and 9 are not accessible through the firewall 3 from the internet 1. As a consequence the SOAP client 4 can only access the SOAP server 5 - or other servers which are not shown in fig. 1 - over the internet 1 but can not access applications provided on servers 8 or 9 in the company intranet 2. This is a serious disadvantage for offering a variety of applications to customers, in particular if existing CORBA and EJB systems are available.

The invention aims to provide a solution for the above described prior art problem and provides an improved method for coupling of different types of client and server objects as well as a corresponding computer system, methods and computer program products as claimed in the independent claims.

The invention couples a client of a first object type, for example a SOAP client, to a server of a second object type, for example a CORBA or a EJB server. The coupling can be achieved over the internet, intranet and firewalls.

In a preferred embodiment this is accomplished by providing a bridge for coupling server and client components of different object types to provide a communication path between the respective client and server via the bridge. The bridge contains a mapping component for mapping of objects of a first type to objects of a second type, for example mapping of SOAP objects to CORBA or EJB objects and vice versa.

The invention is advantageous in that it removes the requirement for using stubs and skeletons for SOAP generated objects by reusing the already existing client stubs of the CORBA and EJB

servers and by creating a generic meta program to locate the initial contexts and homes by accessing pre-configured information. The SOAP based clients must specify the object name, the method name and the parameters.

In addition, the uniform resource locator of the bridge is published together with CORBA and EJB application capabilities to a web directory. Web service requestors - after determining that their "real" service provider can only be accessed by CORBA and EJB protocols - can access the bridge. The bridge then passes on the request "in proxy" to the actual CORBA or EJB service. One application is to store the uniform resource locator of the bridge in a yellow pages type server.

This way the invention realizes distributed application services where parts of the application can be distributed over the internet and intranet being separated by firewalls. This is of particular importance for usage of existing applications which are not firewall capable in such distributed environments.

In the following, the invention is explained in more detail with respect to the drawing in which

Fig. 1 is a prior art computer system having server and clients of different object types,

Fig. 2 is a computer system having an embodiment of the bridge of the invention,

Fig. 3 is the computer system of fig. 2 having a yellow pages server for locating an application service in the internet or intranet, and

Fig. 4 is a flow chart of an embodiment of the invention for coupling objects of different types.

Fig. 2 shows a computer system - using like new numerals for like components as in fig. 1 - having a SOAP client 4 and a SOAP server 5 that are coupled by internet 1. The bridge 10 is contained in the intranet 2, which is coupled to the internet 1 via firewall 18. Intranet 2 further contains CORBA client 6 and EJB client 7 that are coupled to CORBA server 8 and EJB server 9, respectively.

The bridge 10 contains a SOAP server component 11 and a SOAP component client 12. The SOAP server component 11 establishes a communication path between the SOAP client 4 and bridge 10 via the internet 1, whereas the SOAP client 12 serves to establish a communication path between the bridge 10 and the SOAP server 5 via the internet 1 - depending on the particular application.

The bridge 10 further comprises an EJB client component 13 and an EJB server component 14. The EJB client component 13 establishes a communication path between the bridge 10 and the EJB server 9, whereas the EJB server component 14 establishes a communication path between the EJB client 7 and the bridge 10 via the intranet 2. Likewise, the bridge 10 contains CORBA client component 15 and CORBA server component 16. The CORBA client 15 establishes a communication path between the bridge 10 and CORBA server 8, whereas the CORBA server component 16 establishes a communication path between CORBA client 6 and the bridge 10.

The bridge 10 further comprises a mapping component 17 for mapping of an object of a first type to an object of a second type and mapping of corresponding method requests and

parameters. For example the mapping component 17 can map a SOAP based method request to a CORBA or EJB method call at run-time. Likewise the mapping component 17 can map CORBA or EJB method calls to SOAP based method requests.

The bridge 10 has a uniform resource locator as well as the other server and client entities as depicted in fig. 2.

The usual operation of SOAP client 4 is to access the SOAP server 5 with an application request via the internet 1. In case the desired application is not a SOAP application but - in the example considered here - a CORBA or EJB application running on a server of intranet 2, the SOAP client 4 must first access the bridge 10.

This is done for example by providing the uniform resource locator of the bridge 10 to the SOAP client 4. This access request of SOAP client 4 also includes a parameter for indicating the object type of the desired application and further parameters for input into the application on the remote server.

The access request of SOAP client 4 based on the uniform resource locator of bridge 10 invokes the SOAP server component 11, which appears to SOAP client 4 to be a usual SOAP server. Therefore the operation of SOAP client 4 does not need to be modified in any respect. Based on the object type of the desired application a corresponding client component of that object type is selected by the bridge 10.

For example, assume that SOAP client 4 desires access to a CORBA type application on server 8. This is indicated by the SOAP client 4 to the bridge 10 in its access request by specifying

the object type of the application. Based on this indication of the object type, the CORBA client component 15 is selected and activated by the bridge 10.

The CORBA client component 15 appears to the CORBA server 8 to be a usual CORBA client; therefore the operation of the CORBA server 8 does not need to be changed in any respect for integration into the computer system of fig. 2. This way communication paths are established between the SOAP client 4 and the SOAP server component 11 as well between the CORBA client component 15 and the CORBA server 8.

A SOAP based method request of SOAP client 4 is received by bridge 10 by means of its SOAP server component 11. From there it is inputted into the mapping component 17. The mapping component 17 provides a CORBA method call having the same functionality as the inputted SOAP based method request. This CORBA method call is inputted into the CORBA client component 15 from where it is transmitted to the CORBA server 8.

This invokes the desired application on CORBA server 8. The parameter data provided by the SOAP client 4 is inputted into the application of CORBA server 8. For example, the application of CORBA server 8 can be a financial services application, such as portfolio management or financial planning.

In this case the input parameters provided by the SOAP client 4 can be stock market data or the like on which the analysis of the application is based. These input parameters for the application program are transmitted from the SOAP client 4 to the SOAP server 11 and to the CORBA server 8 via mapping component 17 and CORBA client component 15.

Fig. 3 shows a further preferred embodiment having an additional yellow pages type server 19. The server 19 has a data base which contains a number of available application services and corresponding object types of the application services as well as the uniform resource locator of an application service. The database can be searched based on key words, such as by a Boolean key word search. The SOAP client 5 can connect to the server 19 via internet 1 to query server 19 to locate a desired application service.

A corresponding query inputted by the SOAP client 4 into the server 19 results in a hit list from which the SOAP client 4 selects a particular application service. By selecting the application service on this hit list the object type of the application service as well as its uniform resource locator are provided to the SOAP client 4. In addition the data base also contains a uniform resource locator of the bridge 10 for providing a communication path to a desired application service. This uniform resource locator of the bridge 10 is also provided by the server 19 to the SOAP client 4.

For example, assume that the desired financial services application is of the CORBA object type. As the CORBA and SOAP object types differ it is inferred in the SOAP client 4 that access to the required CORBA server 8 is only possible via the bridge 10. Hence SOAP client 4 performs an access operation to the bridge 10 by means of the uniform resource locator of the bridge 10 and provides the object type and the uniform resource locator of the application service to the bridge 10. Again, further input parameters for input into the application service program can be provided by the SOAP client 4.

Based on the object type and the uniform resource locator of the desired application service the bridge establishes a communication link between its CORBA client component 15 (cf. fig. 2) and its CORBA server 8. This way the SOAP client can communicate with the CORBA server 8 as explained in detail with respect to fig. 2.

Fig. 4 shows a preferred embodiment of the method of the invention. In step 20 the SOAP client performs an access operation to a yellow pages type server to identify a desired application. In step 21 the SOAP client queries the yellow pages server to find the URL of the desired application and its object type. In addition, the URL for a corresponding bridge results from the query for access to the application.

In step 22 it is decided whether the object type of the application is of the type "SOAP".

If this is the case, the SOAP client directly connects to the application service in step 23 by entering the URL of the application service into its web browser program.

If the contrary is the case, the SOAP client obtains the URL of the bridge from the yellow pages server in step 24 to access the bridge by means of this URL in step 25.

In step 26 the bridge is provided with the URL of the desired application service and further parameters, such as data indicative of the object type of the application and input parameters for the application.

In step 27 the bridge creates a communication path to a server containing the desired application based on the URL provided by

the SOAP client requests. After this communication path is created the communication between the SOAP clients and the application server through the bridge can start at step 28.

The communication is established and maintained through the server component and the CORBA client component of the bridge by means of the intermediary mapping component of the bridge.

Likewise, it is possible to establish a communication path between a CORBA client and a SOAP server or between an EJB client and a SOAP server by using the corresponding client and server components of the bridge 10. This way it is possible to build distributed application services relying on a variety of object type applications being distributed over different kinds of networks separated by firewalls.

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
219